Partnerships in Innovation

Serving a Networked Nation

Panel 4: Semantic Technologies

Bertram Ludäscher (Moderator)

Associate Professor
Dept. of Computer Science & Genome Center
University of California, Davis

UC DAVIS
Department of
Computer Science

Fellow
San Diego Supercomputer Center
University of California, San Diego

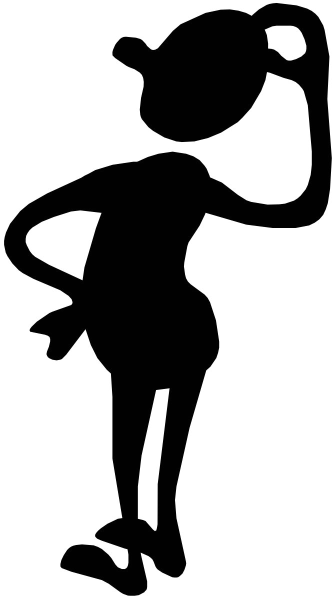
San Diego SDSC
Supercomputer Center

Panel: General Theme

- **What difference can semantic technologies make in digital preservation?**
  - – … in particular, Semantic Web standards and technologies

- **What are the challenges?**

- **But first: What *is* semantics?**



What is Semantics?

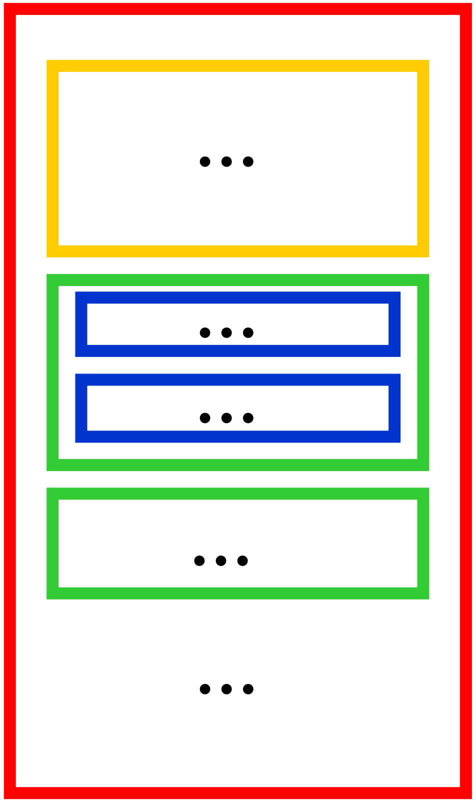
- **Syntax**
  - – how we spell things, e.g.:
    - <a>foo bar<a> (OK) vs. <a] baz </a> (NOT OK)
- **Structure**
  - – how we organize and package things, e.g.:
    - a red “box” (XML element) may contain a yellow box and does contain one ore more green boxes
    - a green box must contain 2 blue boxes, possibly followed by a purple box

```
<red>    → <yellow>?, <green>+
<green>  → <blue>, <blue>, <purple>?
```

XML "Shoebox" Model

Structural Constraint SC

<red> → <yellow>?, <green>+
<green> → <blue>, <blue>, <purple>?



```
<red>
   <yellow> ... </yellow>
   <green>
      <blue> ... </blue>
      <blue> ... </blue>
   </green>
   <green>
      ...
   </green>
</red>
```

Shoebox model **(OK wrt *SC*)**

XML syntax **(OK wrt *SC*)**

What is Semantics?

Semantics

- what we mean (***concepts***) when using certain terms

- defining or describing (new) concepts in relation to other concepts and properties, e.g.:

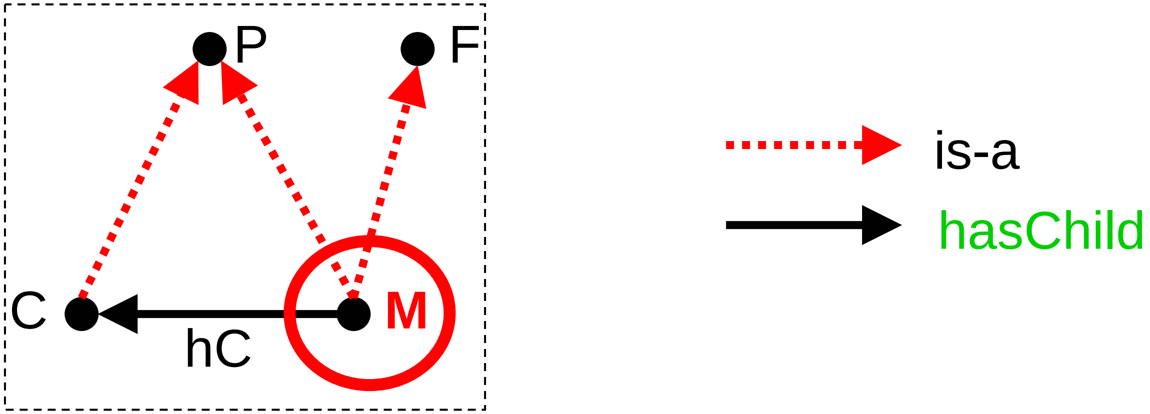
  Mother(x) **:=**
  Person(x) and Female(x) and hasChild(x,y) s.t. Child(y)

- **ontology** as a **semantic reference system** to which we can "register" data & metadata
  - **<red> ~ Mother, <yellow> ~ Spouse, <green> ~ Child**

What the Semantics is …

- **Why not simply <mother> … </mother> ?**
  - XML (DTD/Schema): only "packing instructions"
- **Contrast with capturing (some) semantics :**

Mother(x) → Person(x) and Female(x) and hasChild(x,y) and Child(y)
Child(x) → Person(x)



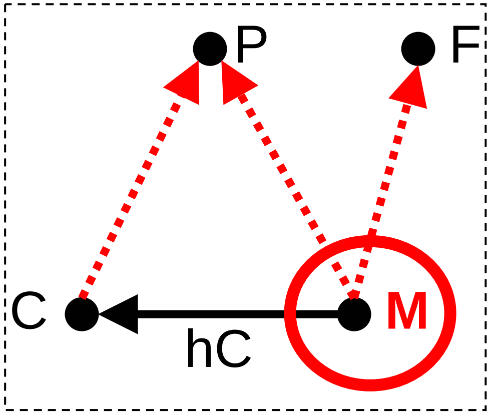
Mother(x) ← Person(x) and Female(x) and hasChild(x,y) s.t. Child(y)

Semantics-Aware (Archival or IR) System

- **Improved Recall**

?- Person(x). *% retrieve also x with Mother(x)*

?- Female(x). *% retrieve also x with Mother(x)*



- **Improved Precision**

?- Mother(x). *% check if Person(x), Female(x) …*

% … qualify

Semantics-Aware (Archival or IR) System

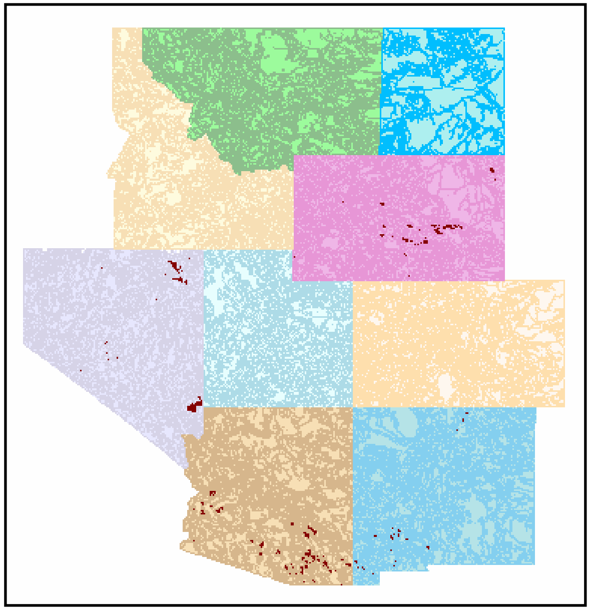
- **Improved Information Quality, Utility, Usability**
  - The **Declaration of Independence** (in Binary)???

  - cf. Hieroglyphs without Rosetta Stone, …
  - ... or having a fine digital copy, encrypted, lost the key

➔ Semantics-aware system adds value

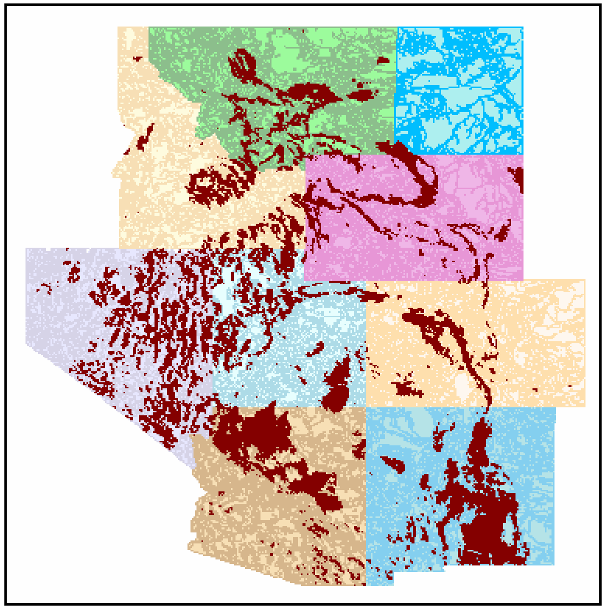
➔ capture information about **content** & **context** in a form amenable to system processing

Example: Semantics-Aware System



?- Paleozoic(x).

***without** ontology*



?- Paleozoic(x).

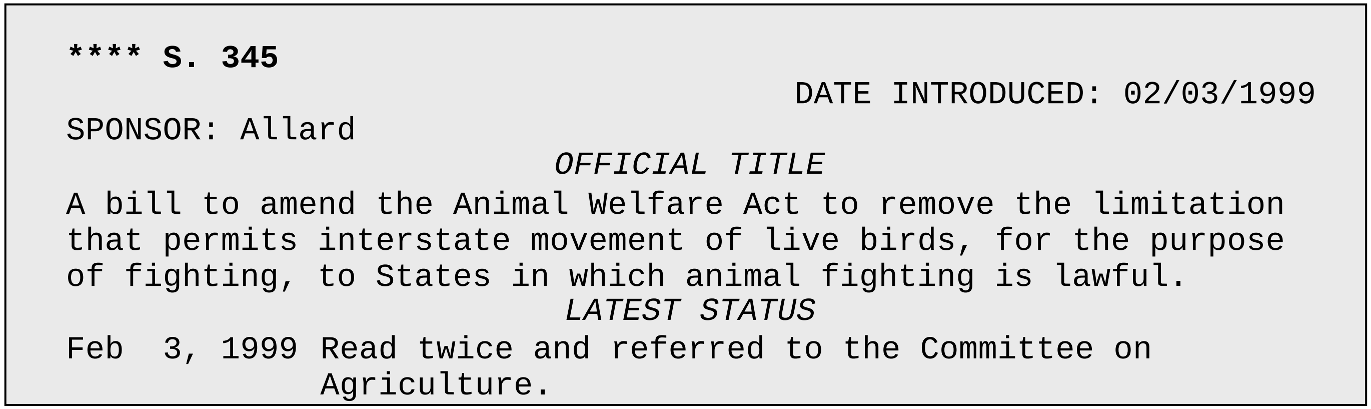
***… with** ontology*:

Cambrium(x)➔ Palezoic(x)

Perm(x) ➔ Palezoic(x), …

- **Value added:**
  - **Concept-level** queries, capturing more **content** & **context**
  - **Improved recall** (more true positives)
  - **Improved precision** (less false positives)

SDSC Case Study: Senate Collection



```
**** S. 345
                                        DATE INTRODUCED: 02/03/1999
SPONSOR: Allard
                         OFFICIAL TITLE
A bill to amend the Animal Welfare Act to remove the limitation
that permits interstate movement of live birds, for the purpose
of fighting, to States in which animal fighting is lawful.
                         LATEST STATUS
Feb  3, 1999 Read twice and referred to the Committee on
             Agriculture.
```

- **Capture syntax, structure, and (some) semantics**
  - add “knowledge packages” (semantic integrity constraints, ontologies) to the archival information package (AIP)
  - additional checks & information at submission and dissemination time

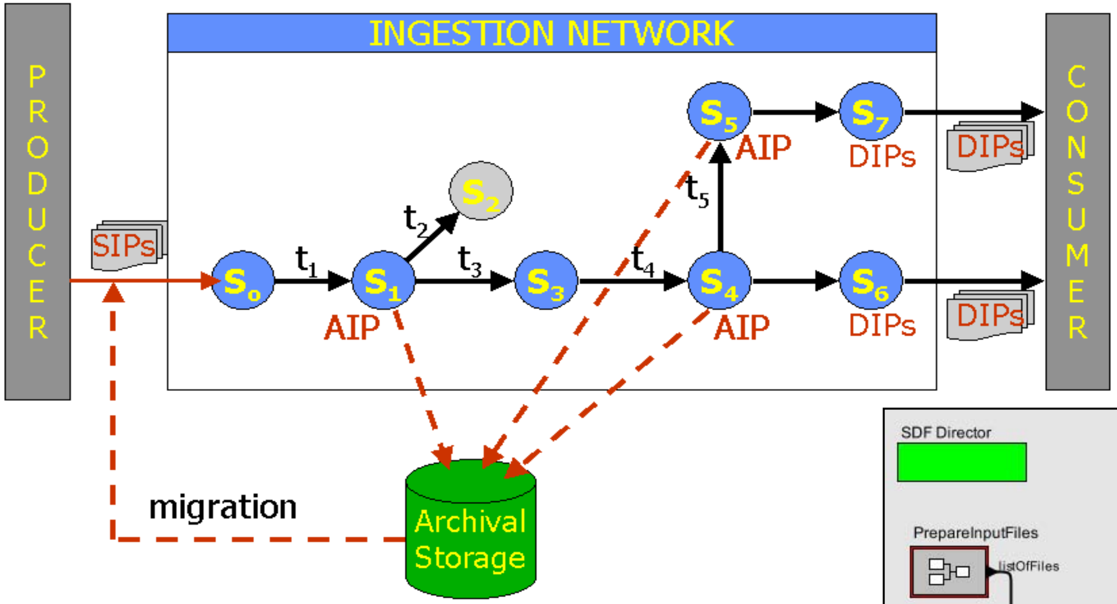
IF sponsor(X), not senator(X) **THEN** ADD(log, missing_senator_info(X))

Self-Describing Data/Metadata/Records

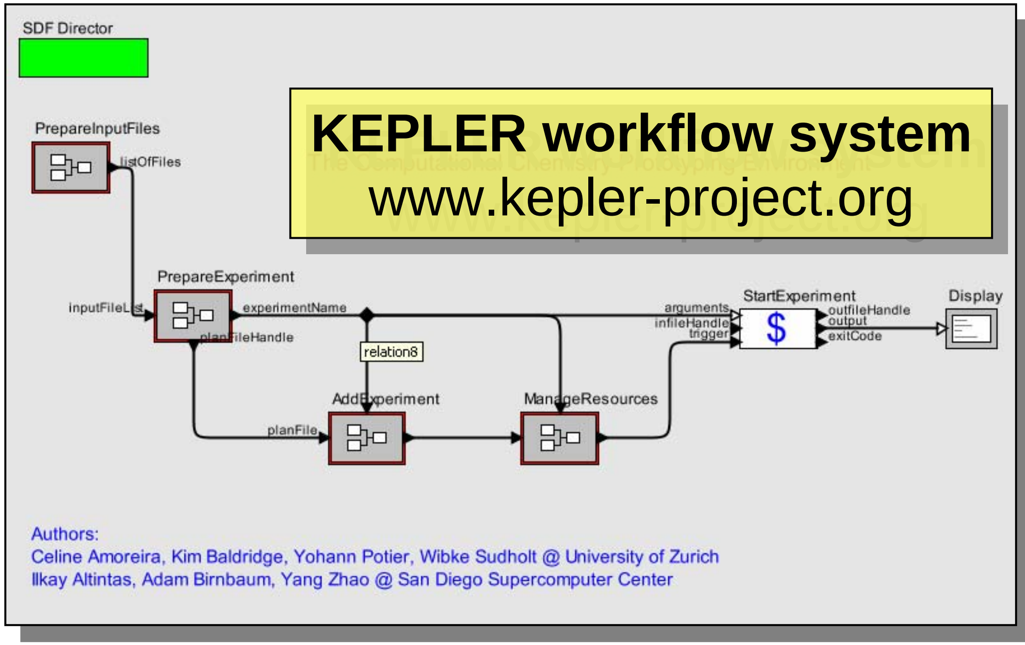
- **XML is “self-describing”:**
  - – **structure** (“packaging instructions”): **YES**
  - – semantics (tag “<mother>”):
    - for human: **YES**, possible (read the Family-ML docu!)
    - for machine (system): **NO**

- **XML+OWL (or other logic) axioms more self-describing:**
  - – structure: **YES** (for human & machine)
  - – **semantics**: **YES** (for human & machine!)

Ingestion Network (“Workflow”)



- Archival processes, submission, ingestion, migration, can be described, captured, and archived as well
- Looking the archivist “over the shoulder”



KEPLER workflow system
www.kepler-project.org

Authors:
Celine Amoreira, Kim Baldridge, Yohann Potier, Wibke Sudholt @ University of Zurich
Ilkay Altintas, Adam Birnbaum, Yang Zhao @ San Diego Supercomputer Center

- Bioinformatics, cheminformatics, ecoinformatics, geoinformatics, … workflows capture data processing and analysis steps and semantics
- use of Semantic Web standards (XML, RDF, OWL, …)

Information Packets may be …

- **Self-contained**
  - – no external links need to be followed
- **Self-describing (for humans)**
  - – no additional info needed; human can understand
- **Self-validating (for machines)**
  - – semantic constraints are packaged as well
  - – machine can “understand” (better: validate)
  - – needs a validation engine (reasoning system)
- **Self-instantiating**
  - – executable, semantically annotated “ingestion workflows” are packaged, too

Semantics Technologies: Summary

Baron von Münchhausen, pulling himself out of the swamp

- Capturing and archiving **semantics** adds value:
  - **additional content and context information**
  - **additional validation at ingestion time**
  - **“smart discovery” at retrieval time**
  - **improved precision and recall**

- The Future:
  - **Self-Instantiating (“bootstrapping”) Semantics-Aware Archives**
  - **“Self-contained ++ semantics ++ workflow processes”**

Semantic Technologies: Panelists

- **Eric Miller**
  - Semantic Web Activity Lead, World Wide Web Consortium (W3C), Research Scientist, AI Lab, MIT

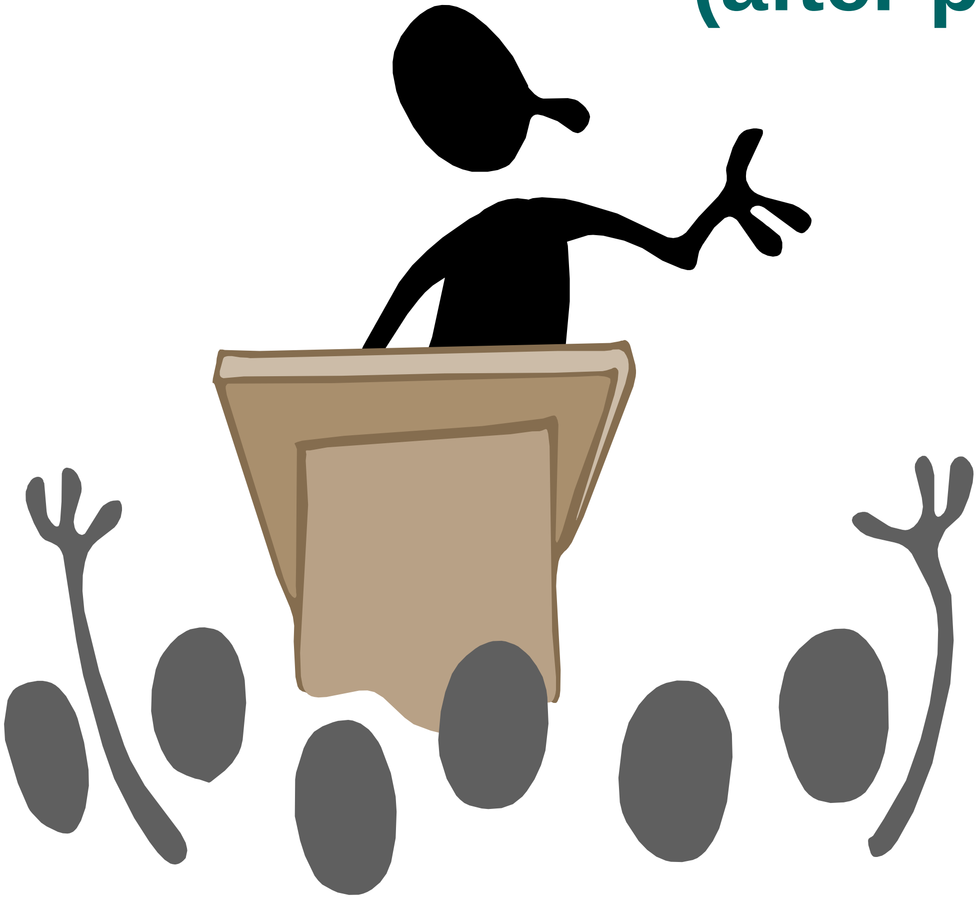
  ➔ **Semantic Web Technology Standards**
- **William Underwood**
  - Principal Research Scientist, Georgia Tech Research Institute, Atlanta; PI of Electronic Records Project (NARA), co-PI InterPARES (long-term preservation of authentic digital record)

  ➔ **Semantic Technologies applied to FOIA Review**
- **John Zimmerman**
  - Kansas City Plant, National Nuclear Security Administration, U.S. Department of Energy

  ➔ **Authenticating Engineering Objects for Digital Preservation**

Q & A

(after panelists’ statements)



Additional Material

In Search of the Semantics

- **Syntactic constraints:**
  - parser can check ***well-formedness*** of document ***D***
- **Structural / schema constraints:**
  - parser can check ***validity*** of ***D*** w.r.t. a ***schema S***
  - “nesting recipe” ***S*** ; also data type checking
- **Semantic constraints:**
  - reasoner can check ***consistency*** of ***D*** w.r.t. a set of ***semantic integrity constraints F***
    - ***F*** can be a set of logic formulas
    - specifically ***F*** can be an ***ontology***

Brief Recall: OAIS Information Packages

- **Information package has multiple components:**

  IP = [DI [PI [CI PDI[PR CON REF FIX]]]

  - IP: Information Package
  - DI: Descriptive Information
  - PI: Packaging Information
  - CI: Content Information
  - PDI: Preservation Description Information
  - PR: Provenance information
  - CON: Context information
  - REF: Reference information
  - FIX: Fixity information

Standards can help at all levels

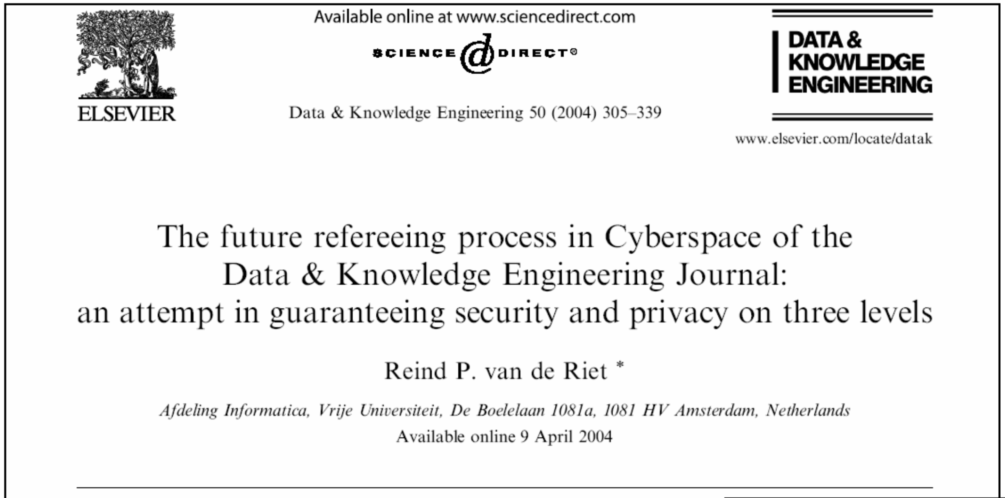
- **Syntax**
  - e.g., use XML
- **Structure**
  - e.g., pick a specific XML Schema or vocabulary
- **Semantics**
  - e.g. pick a specific ontology to capture what the terms of the vocabulary ***mean***
  - **part** of this meaning is accessible to the machine, e.g., whether one concept subsumes another one
  - (NB: need a standard ontology syntax, e.g. OWL)

In Search of the Semantics

- **Further “tagging” of boxes via attributes:**

```
<green creator=“tom” owner=“anne” date=“11/16/04”>
   …
</green>
```

- **But what do the attributes mean?**
  - **owner** of the box or of the content?
  - What **date**? (box vs. content, creation vs. retention,…?)
  - What do **<green>** boxes stand for anyway?
- **Compare these:**
  - <v>56.3</v>
  - <velocity>56.3</velocity>
  - <velocity unit=“miles/hour”>56.3</velocity>
  - … still missing: linking the last one to an ontology!



Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Data & Knowledge Engineering 50 (2004) 305–339

DATA & KNOWLEDGE ENGINEERING

www.elsevier.com/locate/datak

The future refereeing process in Cyberspace of the Data & Knowledge Engineering Journal: an attempt in guaranteeing security and privacy on three levels

Reind P. van de Riet *

Afdeling Informatica, Vrije Universiteit, De Boelelaan 1081a, 1081 HV Amsterdam, Netherlands

Available online 9 April 2004

Abstract

The refereeing process for the Data & Knowledge Engineering Journal as it
Cyberspace is the subject of this paper, in particular security and privacy asp
defined and implemented in the Mokum system; we will show that it complies t
rules on three levels:

1. Highest: at the conceptual level.
2. Middle: at the implementational level.
3. Lowest: at the communicational level.

Keywords: Security and privacy; Cyberspace; Deductive knowledge bases

1. Introduction

To celebrate the forthcoming of the 50th volume of our DKE Journal, t
scientific contribution showing how the refereeing process can be set u
possibilities of Cyberspace are fully exploited. We will emphasize on th

Capturing Workflow Processes in Logic

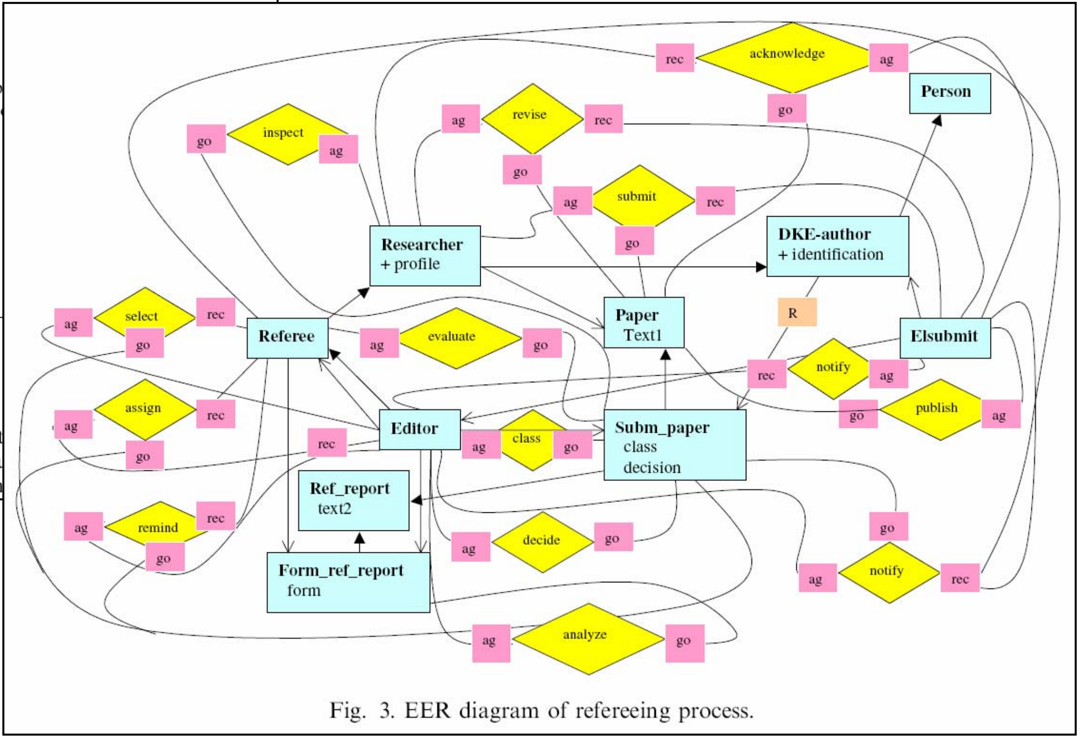
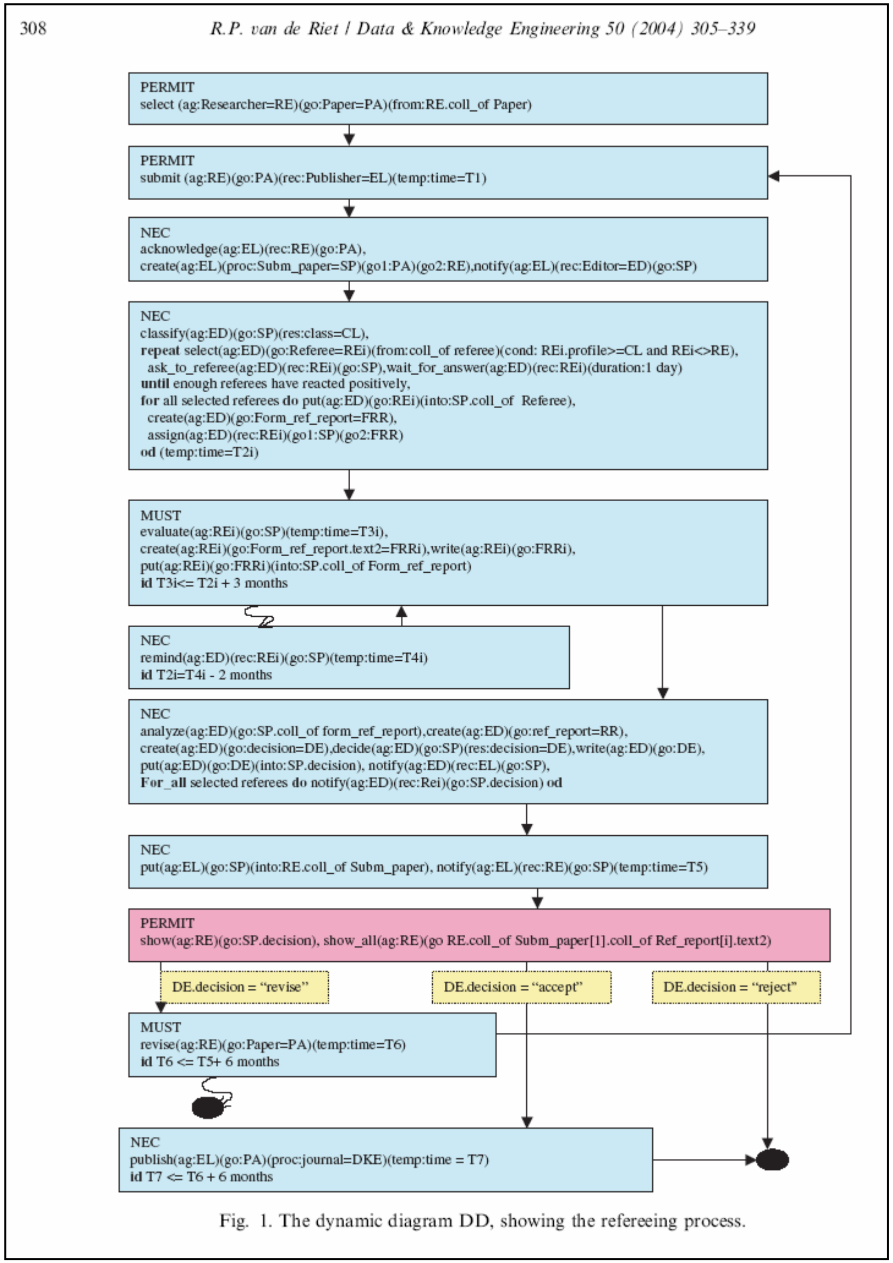


Fig. 3. EER diagram of refereeing process.

Capturing Workflow Processes in Logic



308 R.P. van de Riet / Data & Knowledge Engineering 50 (2004) 305–339

PERMIT
select (ag:Researcher=RE)(go:Paper=PA)(from:RE.coll_of Paper)

PERMIT
submit (ag:RE)(go:PA)(rec:Publisher=EL)(temp:time=T1)

NEC
acknowledge(ag:EL)(rec:RE)(go:PA),
create(ag:EL)(proc:Subm_paper=SP)(go1:PA)(go2:RE),notify(ag:EL)(rec:Editor=ED)(go:SP)

NEC
classify(ag:ED)(go:SP)(res:class=CL),
repeat select(ag:ED)(go:Referee=REi)(from:coll_of referee)(cond: REi.profile>=CL and REi<>RE),
ask_to_referee(ag:ED)(rec:REi)(go:SP),wait_for_answer(ag:ED)(rec:REi)(duration:1 day)
until enough referees have reacted positively,
for all selected referees **do** put(ag:ED)(go:REi)(into:SP.coll_of Referee),
create(ag:ED)(go:Form_ref_report=FRR),
assign(ag:ED)(rec:REi)(go1:SP)(go2:FRR)
od (temp:time=T2i)

MUST
evaluate(ag:REi)(go:SP)(temp:time=T3i),
create(ag:REi)(go:Form_ref_report.text2=FRRi),write(ag:REi)(go:FRRi),
put(ag:REi)(go:FRRi)(into:SP.coll_of Form_ref_report)
id T3i<= T2i + 3 months

NEC
remind(ag:ED)(rec:REi)(go:SP)(temp:time=T4i)
id T2i=T4i - 2 months

NEC
analyze(ag:ED)(go:SP.coll_of form_ref_report),create(ag:ED)(go:ref_report=RR),
create(ag:ED)(go:decision=DE),decide(ag:ED)(go:SP)(res:decision=DE),write(ag:ED)(go:DE),
put(ag:ED)(go:DE)(into:SP.decision), notify(ag:ED)(rec:EL)(go:SP),
For_all selected referees **do** notify(ag:ED)(rec:Rei)(go:SP.decision) **od**

NEC
put(ag:EL)(go:SP)(into:RE.coll_of Subm_paper), notify(ag:EL)(rec:RE)(go:SP)(temp:time=T5)

PERMIT
show(ag:RE)(go:SP.decision), show_all(ag:RE)(go RE.coll_of Subm_paper[1].coll_of Ref_report[i].text2)

DE.decision = “revise”

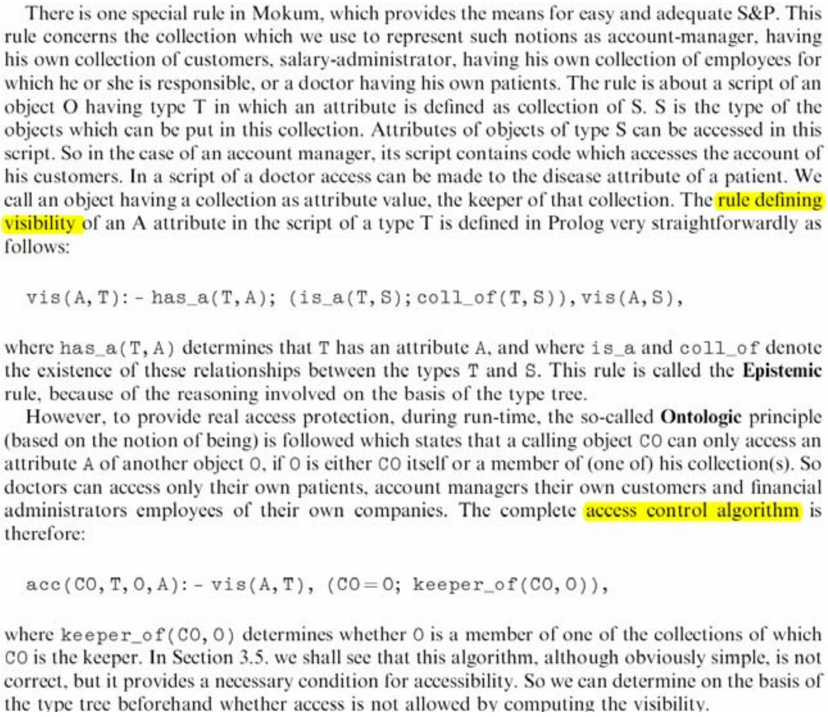
DE.decision = “accept”

DE.decision = “reject”

MUST
revise(ag:RE)(go:Paper=PA)(temp:time=T6)
id T6 <= T5+ 6 months

NEC
publish(ag:EL)(go:PA)(proc:journal=DKE)(temp:time = T7)
id T7 <= T6 + 6 months

Fig. 1. The dynamic diagram DD, showing the refereeing process.



There is one special rule in Mokum, which provides the means for easy and adequate S&P. This rule concerns the collection which we use to represent such notions as account-manager, having his own collection of customers, salary-administrator, having his own collection of employees for which he or she is responsible, or a doctor having his own patients. The rule is about a script of an object O having type T in which an attribute is defined as collection of S. S is the type of the objects which can be put in this collection. Attributes of objects of type S can be accessed in this script. So in the case of an account manager, its script contains code which accesses the account of his customers. In a script of a doctor access can be made to the disease attribute of a patient. We call an object having a collection as attribute value, the keeper of that collection. The rule defining visibility of an A attribute in the script of a type T is defined in Prolog very straightforwardly as follows:

```
vis(A,T):- has_a(T,A); (is_a(T,S);coll_of(T,S)),vis(A,S),
```

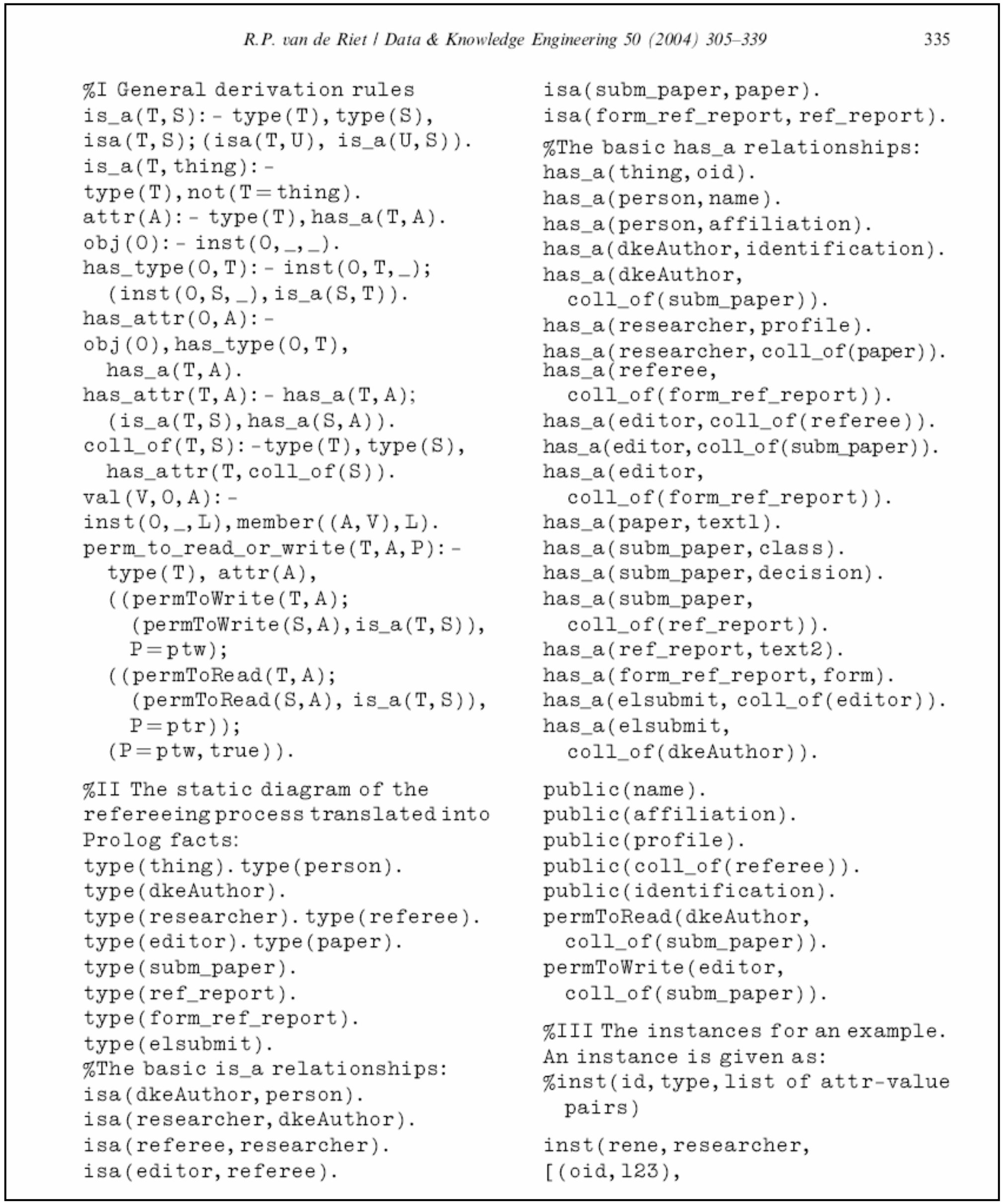
where `has_a(T,A)` determines that `T` has an attribute `A`, and where `is_a` and `coll_of` denote the existence of these relationships between the types `T` and `S`. This rule is called the **Epistemic** rule, because of the reasoning involved on the basis of the type tree.

However, to provide real access protection, during run-time, the so-called **Ontologic** principle (based on the notion of being) is followed which states that a calling object `CO` can only access an attribute `A` of another object `O`, if `O` is either `CO` itself or a member of (one of) his collection(s). So doctors can access only their own patients, account managers their own customers and financial administrators employees of their own companies. The complete access control algorithm is therefore:

```
acc(CO,T,O,A):- vis(A,T), (CO=O; keeper_of(CO,O)),
```

where `keeper_of(CO,O)` determines whether `O` is a member of one of the collections of which `CO` is the keeper. In Section 3.5, we shall see that this algorithm, although obviously simple, is not correct, but it provides a necessary condition for accessibility. So we can determine on the basis of the type tree beforehand whether access is not allowed by computing the visibility.

Capturing Workflow Processes in Logic



```
%I General derivation rules
is_a(T,S):- type(T),type(S),
isa(T,S);(isa(T,U), is_a(U,S)).
is_a(T,thing):-
type(T),not(T=thing).
attr(A):- type(T),has_a(T,A).
obj(O):- inst(O,_,_).
has_type(O,T):- inst(O,T,_);
  (inst(O,S,_),is_a(S,T)).
has_attr(O,A):-
obj(O),has_type(O,T),
  has_a(T,A).
has_attr(T,A):- has_a(T,A);
  (is_a(T,S),has_a(S,A)).
coll_of(T,S):-type(T),type(S),
  has_attr(T,coll_of(S)).
val(V,O,A):-
inst(O,_,L),member((A,V),L).
perm_to_read_or_write(T,A,P):-
  type(T), attr(A),
  ((permToWrite(T,A);
    (permToWrite(S,A),is_a(T,S)),
    P=ptw);
  ((permToRead(T,A);
    (permToRead(S,A), is_a(T,S)),
    P=ptr));
  (P=ptw,true)).

%II The static diagram of the
refereeing process translated into
Prolog facts:
type(thing).type(person).
type(dkeAuthor).
type(researcher).type(referee).
type(editor).type(paper).
type(subm_paper).
type(ref_report).
type(form_ref_report).
type(elsubmit).
%The basic is_a relationships:
isa(dkeAuthor,person).
isa(researcher,dkeAuthor).
isa(referee,researcher).
isa(editor,referee).
```

```
isa(subm_paper,paper).
isa(form_ref_report,ref_report).
%The basic has_a relationships:
has_a(thing,oid).
has_a(person,name).
has_a(person,affiliation).
has_a(dkeAuthor,identification).
has_a(dkeAuthor,
  coll_of(subm_paper)).
has_a(researcher,profile).
has_a(researcher,coll_of(paper)).
has_a(referee,
  coll_of(form_ref_report)).
has_a(editor,coll_of(referee)).
has_a(editor,coll_of(subm_paper)).
has_a(editor,
  coll_of(form_ref_report)).
has_a(paper,text1).
has_a(subm_paper,class).
has_a(subm_paper,decision).
has_a(subm_paper,
  coll_of(ref_report)).
has_a(ref_report,text2).
has_a(form_ref_report,form).
has_a(elsubmit, coll_of(editor)).
has_a(elsubmit,
  coll_of(dkeAuthor)).

public(name).
public(affiliation).
public(profile).
public(coll_of(referee)).
public(identification).
permToRead(dkeAuthor,
  coll_of(subm_paper)).
permToWrite(editor,
  coll_of(subm_paper)).
%III The instances for an example.
An instance is given as:
%inst(id,type,list of attr-value
  pairs)
inst(rene,researcher,
[(oid,123),
```